A Game Theoretic Approach to Efficient Power Management in Sensor Networks[*]

Enrique Campos-Nañez — George Washington University
Alfredo Garcia — University of Virginia
Chenyang Li — University of Virginia

March 20, 2006 (revised August 8, 2006)

Abstract

Wireless sensor networks pose numerous fundamental coordination problems. For instance, in a number of application domains including homeland security, environmental monitoring and surveillance for military operations, a network's ability to efficiently manage power consumption is extremely critical as direct user intervention after initial deployment is severely limited. In these settings, limited battery life gives rise to the basic coordination problem of maintaining coverage while maximizing the network's lifetime. In this paper, we propose a distributed scheme for efficient power management in sensor networks that is guaranteed to identify *suboptimal* topologies in an *on-line* fashion. Our scheme is based upon a general (game-theoretic) mathematical structure that induces a natural mapping between the informational layer and the physical layer. We provide sufficient conditions for the convergence of the algorithm to a pure Nash equilibrium and characterize the performance of the algorithm in terms of coverage. We also present encouraging performance results on a *MicaZ* testbed as well as on large-scale topologies (obtained via simulation).

Keywords: Sensor networks, game theory, distributed algorithms, power management.

[*]We would like to thank Rick Jones and Evan Tschannen for their superb assistance in developing the testbed.

1 Introduction

Sensor networks are evolving into ever more complex arrays of spatially distributed nodes that communicate over wireless channels for collaborative information processing. In light of limited battery life and the complex logistics associated with battery replacement on a large-scale sensor network, efficient power management is a highly desirable feature. In a sensible power management scheme, a number of sensors in a highly dense area would be operated in a 'sleep' mode, while a few designated sensors will be in 'active' sensing mode. This kind of operational strategy may compromise overall network sensing capability. Therefore, finding an optimal trade-off between energy usage and sensing capability constitutes a reasonable objective to aim for, when designing schemes for efficient power management.

Although centralized schemes for energy efficient coverage in sensor networks have been developed (see for instance, [1], [2], [3], [18] , [17] and [11]), they suffer from major drawbacks. Typically, centralization requires substantial communication overhead within a hierarchical architecture designed ex-ante and customized for a given set of contingencies. Communication overhead compromises scalability, while designs and/or plans obtained offline, are not reactive to unspecified contingencies. Recently, a number of distributed schemes for efficient power management in sensor networks have also been proposed (see for instance, [4], [6], [8], [9], [21], [19] and [20]). However, these solutions generally have an ad-hoc flavor as they are often inspired by heuristic arguments that typically work well for very specific scenarios but lack more general theoretical support for their performance. In this paper, we propose a distributed scheme for efficient power management in sensor networks that is guaranteed to identify *suboptimal* topologies in an *on-line* fashion. Our scheme is based upon a general (game-theoretic) mathematical structure that induces a natural mapping between the informational layer and the physical layer.

The structure of this paper is as follows: in Section 2 we formally discuss the game theoretic approach to implementing efficient power management policies in sensor networks. The basic distributed algorithms are then presented. In Section 3, we discuss the convergence of the computationally simplest version of the algorithms presented. In Section 4, the extension to cluster formation and duty cycles is presented. Finally, Section 5 discusses the details of our implementation in MicaZ motes and the computational experience for large-scale scenarios.

2 A Game Theoretic Approach to Efficient Power Management in Sensor Networks

Decentralized and/or distributed control systems can be viewed in mathematical terms as 'games of identical interests' in which 'player' status is attributed to systems components with decision-making or control authority. For example, in Garcia et. al. [7], vehicles traversing a road network are artificially assumed to care for a system-wide criterion (i.e., the average trip time experienced in the network). This assumption is clearly not realistic from a *descriptive* standpoint, since drivers are only concerned with their own trip time. From a *prescriptive* standpoint though, this assumption lends itself well for the construction of a reasonable model for *collaborative decision making* under limited *bilateral communication.* Restricting bilateral communication is a necessary condition for the scalability of any control scheme in a large-scale sensor network. Limited bilateral communication proscribes bargaining and/or the formation of coalitions, which are the subject of study of cooperative game theory. Instead, the paradigm of independent decision making and limited communication fits perfectly into the framework of non-cooperative game theory, albeit, in a game where all players share the same objective.

2.1 Mathematical Framework

Consider a set of n spatially distributed sensors. We assume that for all sensors, the radius of sensor coverage r is no greater than half the communication radius r_c (i.e. $2r < r_c$). Typically, sensing consumes energy at a higher rate than communications. In order to save energy, we want to find a minimal sensor coverage configuration. Let us denote by a a possible sensor operational configuration, i.e. $a = (a_1, a_2, \ldots, a_n)$ where $a_i \in \{0, 1\}$, $i \in \{1, 2, \ldots, n\}$ corresponds to sensor i's decision to turn 'on' (1) or 'off' (0) its sensing capabilities. We define a coverage function $K(a)$ as follows

$$K(a) = \sum_{i=1}^{n} \mathbf{1}_i(a)$$

where

$$\mathbf{1}_i(a) = \begin{cases} 1 & \text{sensor } i \text{ is covered under action profile } a \\ 0 & \text{otherwise} \end{cases}$$

and a sensor is considered covered if *all points within its sensing radius* are covered by the sensor itself or by other active sensors. To quantify the trade-off between energy usage and sensing capability we define a payoff function

$$U(a) = K(a) - c\sum_{i=1}^{n} a_i,$$

where $c \in (0, 1)$ is the 'cost' of turning on a sensor. An optimal sensor coverage configuration is the solution to

$$\max_{a \in A}\{U(a)\} \tag{1}$$

where $A = \{0, 1\}^n$. This problem can be solved in a centralized and off-line mode for reasonable sized instances of the problem, but our interest lies in developing a *distributed* algorithm to obtain suboptimal solutions in an *online* fashion for large-scale settings.

2.2 A Game Theoretic Formulation

We consider sensor i to be an autonomous agent, i.e. a 'player', with capability of selecting his actions $a_i \in A_i = \{0, 1\}$. Given an action profile $a = (a_1, \ldots, a_n)$, we define a_{-i} be the action profile of all sensors *but* sensor i; i.e. $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$. Accordingly, let $A_{-i} = \prod_{j \neq i} A_j$, the set of all possible joint actions by player other than i. All players try to maximize their common payoff, i.e., given an action profile a_{-i}, sensor i's set of best replies is

$$B_i(a_{-i}) = \arg\max_{\alpha \in A_i} U(\alpha, a_{-i}).$$

A profile $a^* = (a_i^*, a_{-i}^*)$ is called a Nash equilibrium if

$$U(a_i^*, a_{-i}^*) \geq U(a, a_{-i}^*), \quad \forall a \in A_i,$$

or equivalently if $a_i^* \in B_i(a_{-i}^*)$. Note that by construction of this game of 'identical interests', every optimal solution of the optimization problem in (1) is a Nash equilibrium. While the converse is not true in general, i.e. some Nash equilibria may not be optimal, they do ensure full coverage as we shall explain in what follows.

2.2.1 Local Test.

Let D_i be sensor i's area of coverage, and let $N(i) = \{j \mid D_i \cap D_j \neq \emptyset\}$, i.e., the set of neighbors of sensor i. Note that sensor i can select its best response to an action profile a_{-i} by gathering information in neighborhood $N(i)$ since the difference between the payoffs is given by

$$U(1, a_{-i}) - U(0, a_{-i}) = K_i(1, a_{-i}) - c - K_i(0, a_{-i}), \tag{2}$$

where $K_i(a_i, a_{-i})$ is a local version of the coverage function $K(a_i, a_{-i})$, i.e.

$$K_i(a_i, a_{-i}) = \sum_{j \in N(i) \cup \{i\}} \mathbf{1}_j(a_i, a_{-i})$$

If the difference in Eqn. (2) is positive, sensor i will choose to turn itself 'on'. If the difference is zero, sensor i will choose randomly, and if the difference is negative, the sensor will choose to turn its sensing 'off'.

2.3 Joint-Strategy Fictitious Play JSFP

The basic premise of this paper is that sensors can autonomously reach an approximate solution to problem (1) by dynamically experimenting with changes in topology that are aimed at improving historical performance (as measured by the payoff function U). This 'groping' for optimality can be seen as a form of 'learning' and has been the subject of intense research efforts by game-theorists (see for instance, [5]). In particular, we focus our interest in a class of 'learning' algorithms known as (joint-strategy) fictitious play. In the (joint-strategy) fictitious play algorithm, players select their new actions as the best response to the *empirical frequency of the other players' joint actions.* More specifically, suppose that $\lambda(a)$ is the probability that joint profile a is played. Also, for player i let

$$\lambda_{-i}(a_{-i}) = \sum_{\{b \in A : b_{-i} = a_{-i}\}} \lambda(b),$$

which represents the distribution of joint actions a_{-i} by players other than i. Let us also denote $U(\alpha, \lambda_{-i}) = \sum_{a_{-i} \in A_{-i}} U(\alpha, a_{-i}) \lambda_{-i}(a_{-i})$. Under the (joint-strategy) fictitious play algorithm, player i will find its best response as any action in the set

$$B_i(\lambda) = \arg\max_{\alpha \in A_i} U(\alpha, \lambda_{-i})$$

where $A_{-i} = \prod_{j \neq i} A_j$. Ties are randomly broken. Let $\beta_i(\lambda) \in B_i(\lambda)$ represent this choice.

Joint-Strategy Fictitious Play (JSFP): Let $a^0 \in A$ be any initial action profile, and let $\lambda^0(a) = \mathbf{1}_{\{a=a^0\}}$, be the initial distribution. Then, the algorithm proceeds recursively by

1. *Computing the joint action*

$$a^{t+1} = (\beta_1(\lambda^t), \ldots, \beta_n(\lambda^t)).$$

2. *Updating the empirical frequency according to*

$$\lambda^{t+1} = \lambda^t + \frac{1}{t+1}[\mathbf{1}_{\{a^{t+1}\}} - \lambda^t].$$

A simple variation of this algorithm consists of keeping track of the last M joint actions ($M > 1$) and allowing sampling as a means to reduce computational effort.

Sampled JSFP with Finite Buffer (SJSFP): Let $a^0 \in A$ be any initial action profile, and let $\lambda_M^0(a) = \mathbf{1}_{\{a=a^0\}}$, be the initial distribution. Then, the algorithm proceeds recursively by

1. *Each player independently draws one sample from λ_M^t and computes a best-reply to the sampled joint action profile:*
$$a^{t+1} = (\beta_1(\lambda_M^t), \ldots, \beta_n(\lambda_M^t))$$
2. *Updating the empirical frequency according to*
$$\lambda_M^{t+1} = \lambda_M^t + \frac{1}{M}[\mathbf{1}_{\{a^{t+1}\}} - \mathbf{1}_{\{a^{t-M}\}}].$$

2.3.1 Local Test & Compact Representation

(Joint-Strategy) Fictitious Play can be *fully distributed*, i.e., each sensor can independently carry out its computation, without central coordination, making use of *local* coverage information. To simplify notation, we define

$$K_i(\alpha, \lambda_{-i}) = E_\lambda[K_i(\alpha, a_{-i})] = \sum_{a_{-i} \in A_{-i}} K_i(\alpha, a_{-i})\lambda(a_{-i}),$$

the expected number of neighbors covered for action α, and let

$$K_i(\lambda) = E_\lambda[K_i(a)] = \sum_{a \in A} K_i(a)\lambda(a),$$

the expected number of neighbors covered for distribution λ. We start by noticing that given a distribution $\lambda(a)$ on the joint action profile a, and based on Eqn. (2), a sensor can select its best reply action by looking at the difference

$$\begin{aligned} E_\lambda[U(1, a_{-i})] - E_\lambda[U(0, a_{-i})] &= \sum_{a_{-i} \in A_{-i}} [K_i(1, a_{-i}) - c - K_i(0, a_{-i})]\,\lambda(a_{-i}) \\ &= K_i(1, \lambda) - c - K_i(0, \lambda). \end{aligned}$$

The last term in the right hand side corresponds to the average number of sensors covered in the neighborhood of sensor i (included). Notice that even though the cardinality $|A_{-i}|$ can be large, sensor i can opt to maintain the frequency with which a number of neighbors k is covered. Formally, sensor i will maintain the frequencies $\underline{\lambda}_i(k)$, and $\bar{\lambda}_i(k)$, this is the number of times k sensors in the neighborhood are covered with, and without the assistance of sensor i (this is, when $a_i = 0$, and $a_i = 1$, respectively). Formally, let

$$\begin{aligned}
\bar{\lambda}_i(k) &= \sum_{a_{-i} \in U_i(k)} \lambda(a_{-i}), \\
\underline{\lambda}_i(k) &= \sum_{a_{-i} \in D_i(k)} \lambda(a_{-i}).
\end{aligned}$$

where $U_i(k) = \{a_{-i} \in A_{-i} : K_i(0, a_{-i}) = k\}$ and $D_i(k) = \{a_{-i} \in A_{-i} : K_i(1, a_{-i}) = k\}$. The feasibility of such compact representation is formalized in the following result:

Lemma 1: *For every distribution $\lambda(a)$, we have*

$$E[U(1, a_{-i})] - E[U(0, a_{-i})] = \sum_{k=0}^{|N(i)|+1} k \left[\underline{\lambda}_i(k) - \bar{\lambda}_i(k)\right] - c.$$

Proof: Let $\lambda \in \Lambda$, be a distribution of the action profile set A. Then, for any sensor i, we have

$$\begin{aligned}
E_\lambda[U(1, a_{-i})] - E_\lambda[U(0, a_{-i})] &= \sum_{a_{-i} \in A_{-i}} \left[K_i(1, a_{-i}) - c - K_i(0, a_{-i})\right] \lambda(a_{-i}) \\
&= \sum_{k=0}^{|N(i)|+1} \Big[\sum_{a_{-i} \in D_i(k)} (K_i(1, a_{-i}) - c)\lambda(a_{-i}) - \sum_{a_{-i} \in U_i(k)} K_i(0, a_{-i})\lambda(a_{-i}) \Big] \\
&= \sum_{k=0}^{|N(i)|+1} k \left[\sum_{a_{-i} \in D_i(k)} \lambda(a_{-i}) - \sum_{a_{-i} \in U_i(k)} \lambda(a_{-i}) \right] - c \\
&= \sum_{k=0}^{|N(i)|+1} k \left[\underline{\lambda}_i(k) - \bar{\lambda}_i(k)\right] - c,
\end{aligned}$$

which proves the result. ■

With a local test, and a compact representation, the joint- strategy fictitious play algorithm can be fully decentralized. This results in the following fully distributed algorithm:

Distributed JSFP *Let $a^0 \in A$ be any initial action profile. Then, the algorithm proceeds recursively by*

1. *Computing the joint action*

$$a_i^{t+1} = \begin{cases} 1 & \text{if } \sum_{k=0}^{|N(i)|+1} k \left[\underline{\lambda}_i^t(k) - \bar{\lambda}_i^t(k)\right] \geq c, \\ 0 & \text{otherwise.} \end{cases}$$

2. *Request neighbor information to compute $K_i(0, a_{-i}^{t+1})$, and $K_i(1, a_{-i}^{t+1})$.*

3. *Updating the empirical frequencies for each $k \in \{0, 1, \ldots, |N(i)| + 1\}$ according to*

$$\begin{aligned} \underline{\lambda}_i^{t+1}(k) &= \underline{\lambda}_i^t(k) + \frac{1}{t+1}[\mathbf{1}_{\{k=K_i(1,a_{-i}^{t+1})\}} - \underline{\lambda}_i^t(k)], \\ \bar{\lambda}_i^{t+1}(k) &= \bar{\lambda}_i^t(k) + \frac{1}{t+1}[\mathbf{1}_{\{k=K_i(0,a_{-i}^{t+1})\}} - \bar{\lambda}_i^t(k)]. \end{aligned}$$

It should be noted here the SJSFP with a finite buffer of size $M > 1$, can also be implemented in a distributed fashion. In this case, $\bar{\lambda}_{i,M}(k) = \sum_{a_{-i} \in U_i(k)} \lambda_M(a_{-i})$ and $\underline{\lambda}_{i,M}(k) = \sum_{a_{-i} \in D_i(k)} \lambda_M(a_{-i})$.Updating is of the form,

$$\begin{aligned} \underline{\lambda}_{i,M}^{t+1}(k) &= \underline{\lambda}_{i,M}^t(k) + \frac{1}{M}[\mathbf{1}_{\{k=K_i(1,a_{-i}^{t+1})\}} - \mathbf{1}_{\{k=K_i(1,a_{-i}^{t-M})\}}], \\ \bar{\lambda}_{i,M}^{t+1}(k) &= \bar{\lambda}_{i,M}^t(k) + \frac{1}{M}[\mathbf{1}_{\{k=K_i(0,a_{-i}^{t+1})\}} - \mathbf{1}_{\{k=K_i(0,a_{-i}^{t-M})\}}]. \end{aligned}$$

3 SJSFP Convergence to Equilibrium

In this section we present a novel convergence result for our proposed variation to Fictitious play. Learning algorithms based upon Fictitious Play are a widely-studied subject in the literature on learning in non-cooperative games (see [14], [5] and the more recent work reported in [15]). In the fictitious play algorithm, players compute at each step, their best action or 'reply' based on the assumption that other players' actions follow a probability distribution in agreement with the *product* of (empirical) marginal distributions. Convergence in general is not necessarily guaranteed (see Shapley's counterexample in [16]). Monderer and Shapley [13] proved convergence of fictitious play (in relative frequencies) to the set of Nash equilibria in games of identical interests and Lambert et al. [10] have provided an extension to the case where players take a polynomially increasing and independent number of samples from the empirical (marginal) distributions of play. However, in the case where players' decisions are correlated, Monderer and Shapley's result and the derived extension in Lambert et al. [10] *do not apply.*

Joint-strategy fictitious play differs fundamentally from fictitious play. In the latter, each player believes other players decisions are *independent* (even if the history of play shows they are in fact *correlated*). Consequently, players do not react to the system's historical performance. In JSFP, players *do not ignore* correlation since their best replies are computed with respect to the joint (empirical) distribution of other player's past actions. Recently, Marden et al. [12] have studied an algorithm that is based upon JSFP where a notion of 'inertia' is introduced: players stick to their previous action with positive probability (*even if this action is no longer optimal*). Based upon earlier work by Young [22], the authors establish convergence (with probability one) to (pure-strategy) Nash equilibria under the rather strong assumption that no player is indifferent between *any two* joint action profiles. Our theorem 1 below is a substantially different result. First, the result holds under a milder assumption on the structure of the function U, namely that all Nash equilibria are strict. Secondly, we abstain from assuming 'inertia' so that in our algorithm, *all* players optimize at *all* times. Finally, we only require *finite* memory of past play.

Theorem 1: *Assuming all Nash equilibria are strict, the sequence* $\{\lambda_M^t : t \geq 0\}$ *generated by Sampled JSFP with Finite Memory converges (with probability 1) to a pure strategy Nash equilibrium.*

Proof: See appendix. ■

Having discussed convergence to Nash equilibrium, the question that follows pertains to the 'quality' of these solutions in regards to problem 1. While the limit points in Theorem 1 only exhibit a certain kind of 'local' optimality in that no sensor can improve upon the system's performance by unilaterally pursuing a different course of action, the empirical evidence (to be discussed in Section 4) is extremely encouraging. Regarding coverage, the following proposition provides a partial characterization:

Proposition 1 *If a^* is a Nash equilibrium then full coverage is guaranteed, i.e.,*

$$K(a^*) = n.$$

Furthermore, the coverage obtained is minimal in the sense that no sensor can turn off without exposing part of the network.

Proof: To see why this is true, suppose that sensor i is not 'on' under this profile, i.e. $a_i^* = 0$, and suppose sensor i is not covered. Then, we can conclude that

$$K_i(1, a_{-i}^*) - K_i(0, a_{-i}^*) \geq 1 > c,$$

which shows that player i could improve by unilaterally changing its actions to $a_i = 1$, contradicting our assumption that a^* was an equilibrium. The fact that the coverage is minimal follows immediately from the definition of Nash equilibrium. ■

4 Coverage & Duty Cycles

In order to extend the life of the network through energy management it is not only necessary to produce a minimal coverage, but to organize sensors into a partition where each set in the partition has coverage of the area. This organization can be an aid to implement a so-called 'duty cycle'. In this section we show that our game theoretic formulation can be easily extended to accommodate this situation, and that the JSFP algorithm can be accordingly modified to solve the problem in a distributed fashion.

In this situation, we want to organize the n sensors in a partition consisting of $L+1$ sets. The first set will consists of sensors that will not participate in the duty cycle, while it is desirable that each of the L remaining sets will by itself provide maximal coverage. To accommodate such decision problem, we can see sensor i as a player that now has an action set $A_i = \{0, 1, 2, \ldots, L\}, i = 1, 2, \ldots, n$; this action will correspond to joining the inactive set ($a_i = 0$), or joining any of the L sets participating in the duty cycle ($a_i \in \{1, 2, \ldots, L\}$). We formalize this decision problem in a fashion similar to the coverage problem introduced earlier, i.e., we want to choose an action profile $a = (a_1, a_2, \ldots, a_n) \in A = \prod_{i=1}^{n} A_i$ to solve the optimization problem

$$\max_{a \in A_1 \times \cdots \times A_n} \quad U(a) = \sum_{\ell=1}^{L} \left[K(a(\ell)) - c \sum_{i=1}^{n} a_i(\ell) \right], \tag{3}$$

where the function K is defined as before, and for an action profile $a = (a_1, \ldots, a_n)$, we define the projected action profile

$$a(\ell) = (a_1(\ell), \ldots, a_n(\ell)),$$

where

$$a_i(\ell) = \begin{cases} 1 & \text{if } a_i = \ell, \\ 0 & \text{otherwise.} \end{cases}$$

As before, we consider sensor i to be an autonomous 'player' who will choose its actions from the set A_i to maximize its (common) payoff function of Eqn. (3). Given an action profile $a_{-i} \in A_{-i} = \prod_{j \neq i} A_j$, sensor i's best reply set is given by

$$B_i(a_{-i}) \in \arg\max_{\alpha \in A_i} U(\alpha, a_{-i}).$$

Sensor i need only use *only local information* to select its best reply to other players' actions. Suppose sensor i is comparing between the actions $a_i = \ell$ and $a_i' = \ell'$, with $0 < \ell, \ell'$. Given an action profile

a_{-i} of players other than a_i, we have that

$$\begin{aligned} U(\ell, a_{-i}) - U(\ell', a_{-i}) &= K_i(1, a_{-i}(\ell)) + K_i(0, a_{-i}(\ell')) - \left[K_i(1, a_{-i}(\ell')) + K_i(0, a_{-i}(\ell))\right] \\ &= K_i(1, a_{-i}(\ell)) - K_i(0, a_{-i}(\ell)) - \left[K_i(1, a_{-i}(\ell')) - K_i(0, a_{-i}(\ell'))\right], \end{aligned}$$

which can be evaluated using local information. Further analysis shows that if sensor i is to join any active set, it should join the set that would observe the maximum coverage improvement. More specifically, the candidate set $\bar{\ell}$ can be selected as

$$\bar{\ell} \in \arg\max_{\ell \in A_i} K_i(1, a_{-i}(\ell)) - K_i(0, a_{-i}(\ell)).$$

Finally, sensor i should compare the decision of joining the set $a_i = \bar{\ell}$ to the decision of not joining any partition ($a_i = 0$), i.e.,

$$U(\bar{\ell}, a_{-i}) - U(0, a_{-i}) = K_i(1, a_{-i}(\bar{\ell})) - c - K_i(0, a_{-i}(\bar{\ell})),$$

which can also be evaluated with local information. Moreover, given a distribution $\lambda(a)$ of the joint actions $a \in A$, sensor i can compute its best response using entirely local information. Regarding coverage, the following proposition provides a partial characterization:

Proposition 2 *Let a^* be a pure strategy Nash equilibrium of the duty cycle problem. Then, we have that for every sensor $i = 1, 2, \ldots, n$:*

1. *The sensor will not be assigned to any active set if and only if its neighborhood is covered on all active sets in the duty cycle, i.e.,*

$$a_i^* = 0 \quad \Leftrightarrow \quad K_i(0, a_{-i}^*(\ell)) = N(i), \qquad \forall \ell = 1, 2, \ldots, L$$

2. *The overall coverage cannot be improved by reassigning a single sensor, i.e.,*

$$a_i^* = \ell \neq 0 \quad \Leftrightarrow \quad K_i(1, a_{-i}^*(\ell)) - K_i(0, \ell') \geq K_i(1, a_{-i}^*(\ell')) - K_i(0, \ell), \quad \forall \ell' = 1, 2, \ldots, L, \ell' \neq \ell$$

Proof: Let a^* be a pure strategy Nash equilibrium, and suppose that $a_i^* = 0$. This implies that for all active set $\ell = 1, 2, \ldots, L$, we have

$$K_i(1, a^*_{-i}(\ell)) - c < K_i(0, a_{-i}(\ell)),$$

which clearly implies that by joining active set ℓ, the coverage is not improved by more than $c < 1$, i.e., coverage in active set ℓ is not improved by adding sensor i to the set. Part 2 of this proposition is an immediate consequence of the definition of a pure strategy Nash equilibrium. ■

5 Implementation

5.1 Developing a Testbed with *MicaZ* motes

We implemented the algorithm on *MicaZ* motes[1] to test its performance. For the development of the testbed, we made a simple modification to the original algorithm. In theory, the algorithm requires checking every coordinate in the area formed by a mote and its neighbors to determine whether or not a mote is covered. We restrict this test to a *finite* number of points in the area under consideration. This approach saves processing power at the cost of accuracy in the estimation of coverage. The implementation proceeded as follows.

- First, the base station sends each mote its 2 dimensional coordinates. After a mote has localized itself, it broadcasts its location information with the mote ID to all its neighbors via a `NeighborMsg`. When a mote hears a `NeighborMsg`, it will add the ID to its neighbor table (if the ID has not been added before) and the corresponding mote is considered within the communication range.

[1]2.4 GHz CPU, Flash memory of 128K bytes, IEEE 802.15.4 compliant, 250 kbps data rate (see www.xbow.com for more details).

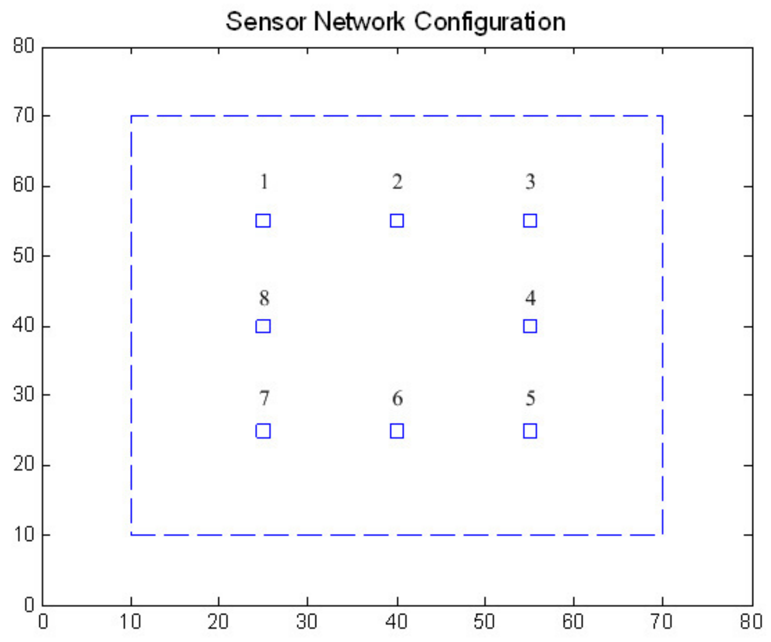

Figure 1: Sensor Network Configuration for Test #1

- After the neighbor discovery is done, the base station floods the network with a 'start coordination' command. All of the motes will receive this message fairly quickly. When a mote receives this message, it starts a timer which periodically initials the algorithm to make a decision about which cluster to join. When the calculation is finished, the mote sends its decision with the stage number to its neighbors to help them make a decision for the next stage. When a mote receives a decision from a neighbor, it updates a joint action array which keeps track of every neighbor's decision for a specific stage. Once a mote has received a decision from every neighbor for a specific stage, it adds that joint action to its memory.

- When the stage number reaches a preset iteration number, the mote will stop the calculation and choose the last decision.

5.1.1 Experimental Results

We set the surveillance area as a 60×60 area. Eight nodes are deployed to that area at location (25,55), (40,55), (55,40), (55,25), (40,25), (25,25) and (25,40) as shown in Figure 1. We assume the communication range and the sensing range are the same for all nodes and $r_c = r_s = 15\sqrt{2}$ so that when turned on, the 4 corner nodes can just cover the edge of the whole area, see Figure 2. We can also see from Figure 2 that corner nodes 1, 3, 5, 7 will have 2 neighbors each and the edge nodes 2, 4, 6 and 8 will have 4 neighbors each.

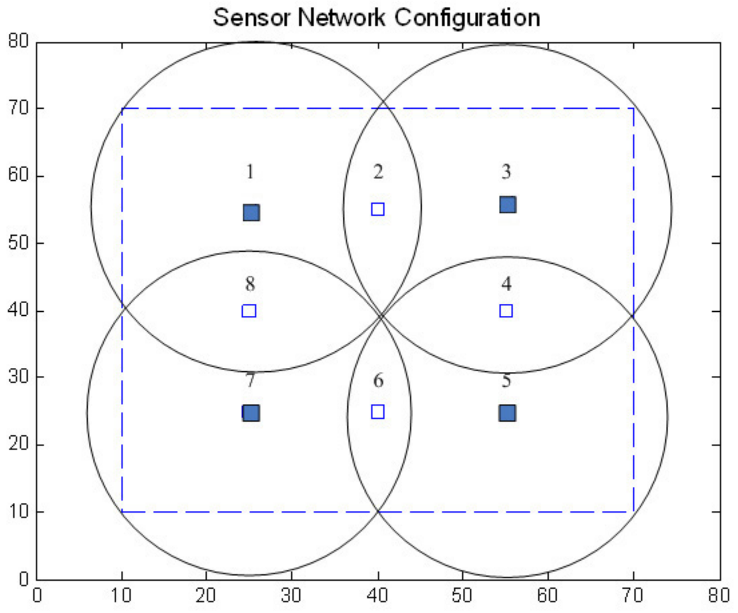

Figure 2: Sensor Network Coverage for Test #1

With two groups, the optimal coverage scheme would be for the 4 corner nodes to form a group (with a 100% coverage) and the remaining 4 edge nodes to form a group (with a 85.42%) coverage). Therefore the optimal average coverage for this scenario is 92.71%.

We set a memory buffer size of 6 and the whole coordination phase will stop when the stage number reaches 20. The results for running the same experiment 200 times (each with a random initial joint decision for the whole network), we got an average coverage of 90.78%, with a 1.82% standard deviation and for 157 out of 200 times, the algorithm converged to the optimal coverage scheme within the fairly limited number of iterations allowed.

In a second test, 25 nodes (i.e., motes) were deployed at 25 non-overlapping grids in the same area as shown in Figure 3. The position for each node within each grid is randomly chosen. This can be viewed as a 'walking' GPS method for localization with deployment error. From the previous experiment, we know that to fully cover the area, we need at least 4 nodes and they must be in the ideal locations. Therefore, 4 to 5 groups with an average group size of 5 to 7 would be very reasonable for this scenario.

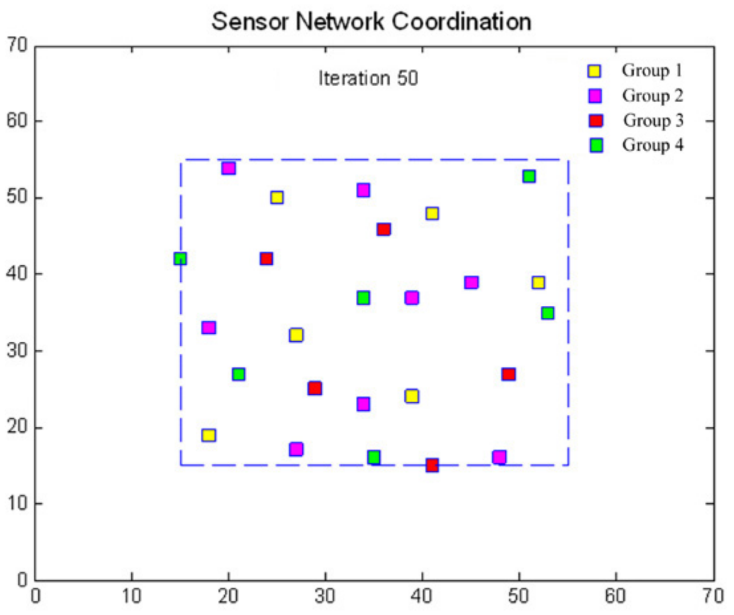

Figure 3: Group Break Down for Test #2: 25 motes in 4 clusters

We ran the experiment 200 times with a memory size of 6 and maximum stage number of 50. With 4 clusters, an average coverage of 97.04% (with a standard deviation of 1.29%) was obtained. For 5 clusters, the average coverage decreases to 91.4%, the standard deviation increases to 1.88%. Evidently,

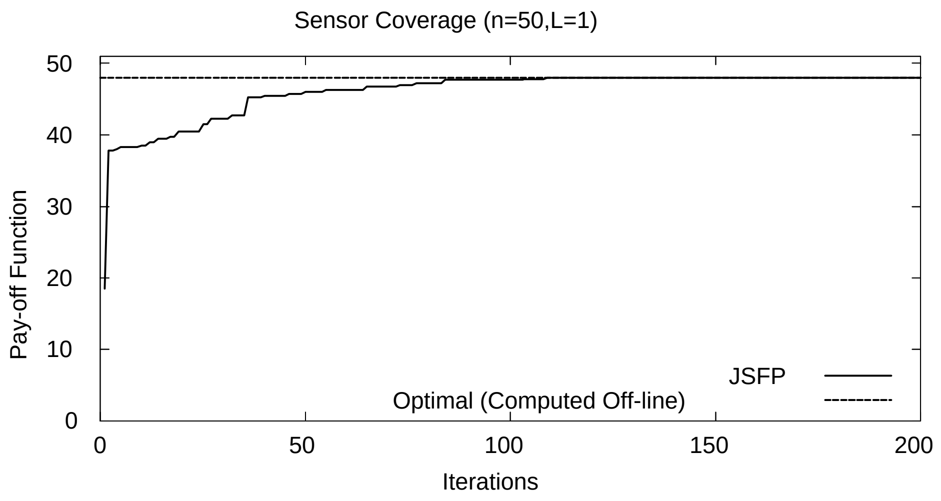

Figure 4: Evolution of Performance for 50 nodes with random locations.

there is a trade-off between the total number of clusters (e.g. the more clusters the more energy is saved) and the coverage that can be achieved. The acceptable level of coverage to total number of groups can be decided by the user.

5.2 Large-Scale Simulation & Optimality Tests

The limit points guaranteed by Theorem 1 exhibit a certain kind of 'local' optimality in that players can not improve upon the system's performance by unilaterally pursuing a different course of action. Evidently, this does not preclude the possibility that two or more players can improve upon the system's performance by jointly deviating from the prescribed course of action. However, our numerical results invariably show that JSFP approximates an optimal solution. For instance, in Figure 4 we show a typical sample path of the evolution of the algorithm's performance for a problem with $n = 50$, for which the optimal solution can be computed off-line with a standard mixed-integer programming solver. In Figure 5 we show the output obtained for a network of $n = 500$ sensors, which are organized into $L = 4$ partitions, compared to the results of a simulated annealing algorithm applied to the same problem.[2] The simulated annealing algorithm was implemented by searching over the neighborhood of a given solution, defined as the set of solutions that differ only in the action of one sensor, and using standard temperature functions. Our simulation runs indicate that the JSFP always improves upon the simulated annealing. We also compared Simulated Annealing and the finite buffer version of JSFP (with the more restricted estimation of coverage using a finite grid). The results are displayed in Table 1.

	Average Coverage	Mean	STD	Max
25 Nodes	Simulated Annealing	98.39%	0.47%	99.17%
	JSFP	97.78%	1.16%	
100 Nodes	Simulated Annealing	98.78%	0.28%	99.26%
	JSFP	98.44%	0.59%	

Table 1: Performance Comparison with 200 runs.

- First, we tried the 25 nodes, 4 clusters, random deployment case we used in Section 5.1.2. We ran both algorithms 200 times on the same network configuration and for our algorithm, we keep a constant memory size of 6 and a maximum stage number of 50.

- As can be seen from Table 1, Simulated Annealing outperforms the finite buffer version of JSFP in both mean and standard deviation of the average group coverage. However, the differences are very small (0.61% and 0.69% respectively). The maximum average group coverage found by Simulated Annealing in the 200 runs was 99.17%, this can be viewed as the optimal average group coverage for this 25 node scenario. This shows that on average, the coverage achieved by our algorithm is only 1.39% less than the best average group coverage that can be obtained.

[2]Movies with sample patha of the algorithm's performance are available at `www.sys.virginia.edu/techreps` under SIE-060006.

- Then we increased the area size by 4 times and deployed 100 nodes over that area. All the other settings remain the same. We see that when the scale is up, both algorithms achieved better performance in mean and standard deviation of average group coverage. Although Simulated Annealing is still leading, the differences are decreasing (0.34% and 0.31% respectively). Also notice that compared with the 25 nodes case, the maximum average group coverage increased from 99.17% to 99.26%. The better performances by both algorithms in the larger scale case can be explained by the 'edge' effects. When the area size is increased by 4 times, the edge areas are only increased by 2 times and the corner areas remain the same. Since the nodes density has not changed, the edge effects are actually decreased, which results in better average group coverage for both algorithms.

These tests indicate that the average group coverage obtained by our algorithm is very close to the optimal solution. Moreover, as the scale is increased the quality of our algorithm's performance increases. Here, it is worth emphasizing that for any centralized approach the computational time required for finding the optimal solution grows exponentially. In contrast, the computational experiments suggest that our distributed approach is not affected by scale.

5.3 Sensitivity Analysis

In Figure 6 to Figure 8, we show the results for running the JSFP algorithm with different nodes densities. We use the same 60m×60m area as in Section 5.1.1 and keep $r_c = r = 15\sqrt{2}$m such that, when turned on, 4 nodes can just cover the whole area. Firstly, 25 nodes are deployed to the area in 25 non-overlapping grids. The position for each node within each grid is randomly chosen. We let L take the values from 4 to 8 and for each chosen value, we run the algorithm 100 times and average the results. The percentage of average coverage at different number of iterations are shown in Figure 6. Then we repeat the same experiment with 36 nodes and 49 nodes deployed to the same area. The

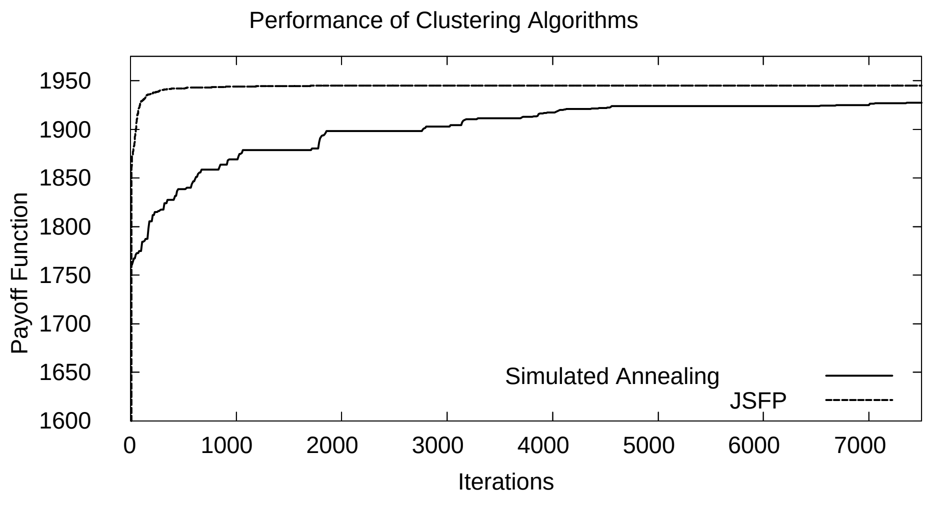

Figure 5: Optimality Test for Large-Scale Problem ($n = 500$ random locations, $L = 4$)

average numbers of neighbors for each sensor in the 3 scenarios are 6, 9 and 13 respectively.

From Figure 6 to Figure 8, we see that given a density of nodes increasing the number of clusters will inevitably decrease the average coverage. When the nodes density is low (25 nodes), a relatively small number of iterations (30) is enough for the algorithm to quickly settle into the best performance. However, with higher densities, neighborhoods are more populated and sensors need more iterations to approximate the best performance.

Figure 6 to Figure 8 also illustrate a way for the decision maker to select L and the number of iterations needed. As mentioned before, L is a user-preset parameter, which represents a tradeoff between the surveillance coverage and system lifetime. Knowing the surveillance area size and number of sensors to drop, the decision maker can, prior to deployment, select a value of L that achieves the desired tradeoff bewteen surveillance coverage and system life-time requirements by conducting an extensive simulation analysis.

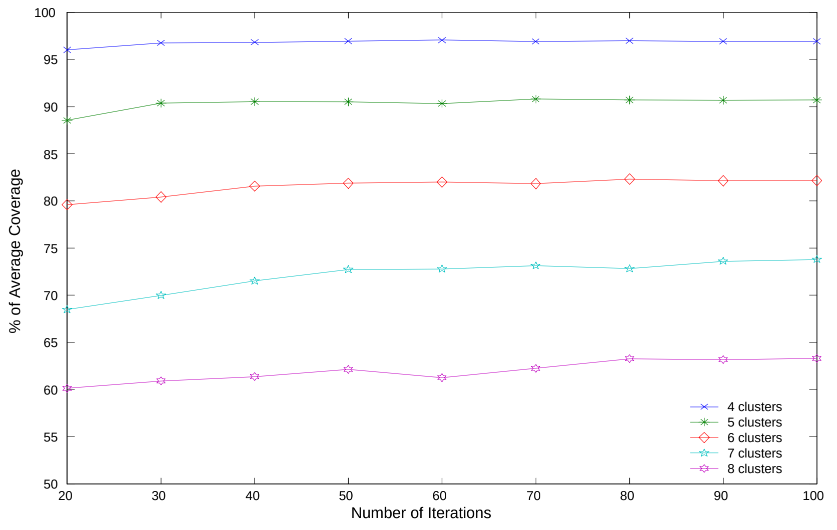

Figure 6: 25 randomly deployed sensors with different partition schemes

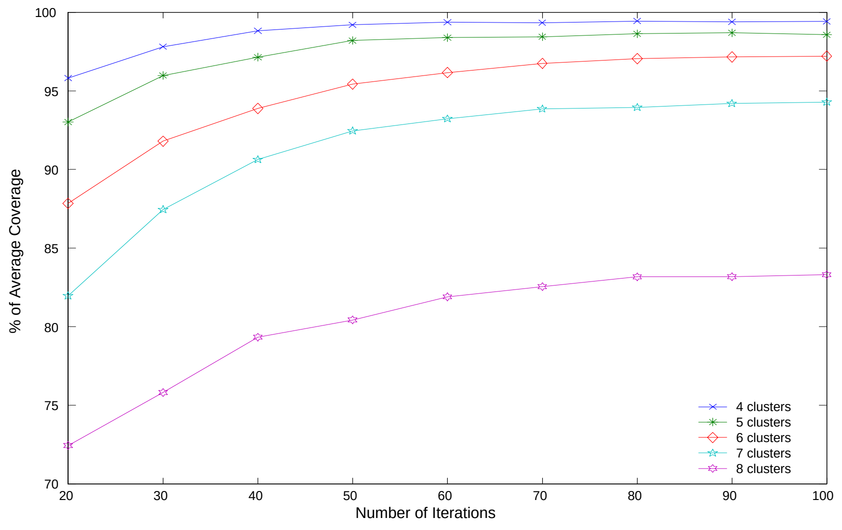

Figure 7: 36 randomly deployed sensors with different partition schemes

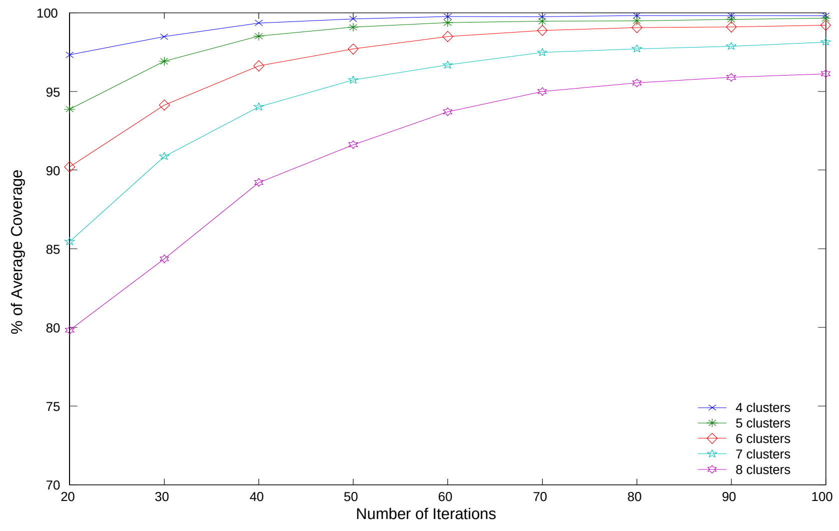

Figure 8: 49 randomly deployed sensors with different partition schemes

5.4 Comparison with Other Centralized Algorithm for Coverage

Cardei et al proposed in [2] a MC-MIP (Maximum Covers using Mixed Integer Programming) algorithm for maximum disjoint set covers problem. It is a centralized approach looking for maximum number of disjoint sets each of which can cover 100% of the target points. Since this algorithm only finds disjoint sets that gives 100% coverage, in order to make a comparison with our SJSFP algorithm, we extend the MC-MIP algorithm in the following way. First, we select 100 points that evenly distributed in the 60mx60m surveillance area as the target points. Then we run the MC-MIP algorithm to find the disjoint sets that give 100% coverage. After that, we remove the sensors that belong to the sets identified (as well as the target points that can only be covered by these sensors) and run the algorithm again. The algorithm stops when no sensors and target points remain. In Table 2, we present sample results of MC-MIP for 100 targets to be covered by 49 randomly deployed sensors.

Cluster #	Number of Sensors	Coverage %
1	5	100
2	6	100
3	6	100
4	7	100
5	6	96.69
6	7	93.39
7	7	82.64
8	4	61.98
9	1	31.34

Table 2: MC-MIP Performance

Comparing the performance of MC-MIP and SJSFP after 100 samples yields the following average coverage:

Number of Clusters	MC-MIP	SJSFP
4	100	99.88
5	99.34	99.41
6	98.35	98.89
7	96.10	97.53
8	91.84	95.89
9	85.12	90.94

Table 3: MC-MIP vs SJSFP

We tested both algorithms for different densities on the same setup (25 and 36 nodes). In most of the cases, with the same number of clusters, the average coverage achieved by JSFP is higher than that of the extended MC-MIP, especially when the cluster size is small. We have to admit that MC-MIP algorithm was originally designed for 100% coverage and that this comparisons are based upon the above described extension of the algorithm.

6 Conclusions

Sensor networks are evolving into ever more complex arrays of spatially distributed nodes that communicate over wireless channels for collaborative information processing. In many application domains (homeland security, environmental monitoring & control and disaster management) a network's ability to efficiently manage power consumption in autonomous fashion is extremely critical, as direct user intervention after initial deployment is severely limited by exogenous factors. This feature precludes the use of centralized schemes that require substantial communication overhead. Communication overhead compromises scalability and centralized hierarchical schemes are limited in their ability to react to unspecified contingencies. Consequently, a number of schemes for distributed control in sensor networks have been recently proposed to address important operational issues such as energy management and adaptive routing. Nonetheless, these solutions have generally been supported by (ad-hoc) heuristic arguments, that typically work well for very specific scenarios but lack general theoretical support for their performance. In this paper, we have presented a novel *distributed* scheme for *on-line* power management in sensor networks that is guaranteed to identify *suboptimal* solutions in an *on-line* fashion. Our setup provides a game theoretic mathematical framework for distributed control in sensor networks. This novel feature induces a natural mapping between the informational layer and the physical layer and is likely to motivate new synergistic technologies. We have provided sufficient conditions for the convergence of the algorithm to a pure Nash equilibrium and have characterized the performance of the algorithm in terms of coverage. Performance results on a *MicaZ* testbed as well as on large-scale topologies are extremely encouraging.

7 Appendix

7.1 Proof of Theorem 1

Consider a sample path $\{a^t : t \geq 1\}$. Let us define the set E as follows:

$$a \in E \Longleftrightarrow a^t = a \text{ infinitely often}$$

Note that at every iteration, with probability $(\frac{1}{M})^n$, all players will compute their best reply with respect to the *same* entry in the buffer. Hence, for all $a \in E$, there exists $a' \in E$ such that $a \in B(a')$. Let $a^* \in A$ be defined as

$$a^* \in \arg\min_{a \in E} \{U(a)\} \tag{A.1}$$

We claim a^* is a Nash equilibrium. By contradiction, let us suppose there exists a deviation from a^*, say $(\tilde{a}_i, a^*_{-i})$ such that

$$U(\tilde{a}_i, a^*_{-i}) = \min_{a_i \in A_i} U(a_i, a^*_{-i}) < U(a^*) \tag{A.2}$$

Let $a' \in E$ such that $a^* \in B(a')$. Note that every time a' is played, with probability $(\frac{1}{M})^n$, a^* will be played afterwards. Having $\{a', a^*\}$ as the last two entries in the buffer implies that $(\tilde{a}_i, a^*_{-i})$ will be played with some positive probability (i.e. all players except player i compute their best reply with respect to a' and player i computes its best reply with respect to a^*). In summary, every time a' is played there is a positive probability that $(\tilde{a}_i, a^*_{-i})$ is played after two iterations. Since a' is played infinitely often, so is $(\tilde{a}_i, a^*_{-i})$. That is, $(\tilde{a}_i, a^*_{-i}) \in E$. This fact coupled with $A.2$ constitute a contradiction to A.1. Hence, a^* is a Nash equilibrium.

Let $\{a^{t(k)} : k \geq 1\}$ be the subsequence such that $a^{t(k)} = a^*$. Without loss of generality assume $t(k) - t(k-1) \geq M$ for $k \geq 1$. In words, if points along the subsequence are at times fewer than M periods away, we work with a sub-subsequence that has property that the distance between succesive times in the subsequence greater than M. Let us denote by C_k the event in which the process converges to a^*, after a sequence of best replies equal to a^* of length M occurs, starting at time $t(k)$:

$$C_k = \bigcap_{l=1}^{M} \{a^{t(k)+l} = a^*\}$$

Note that

$$P(C_{k+1}|\, C_k) = 1 \quad \text{and} \quad P(C_{k+1}|\, C_k^c) \geq (\tfrac{1}{M})^{n\times M}$$

Thus,

$$\begin{aligned} P(\bigcap_{k\geq 1} C_k^c) &= \lim_{T\to\infty} P(\bigcap_{k=1}^{T} C_k^c) \\ &= \lim_{T\to\infty} \prod_{k=1}^{T-1} (1 - P(C_{k+1}|\, C_k^c)) \\ &\leq \lim_{T\to\infty} (1 - (\tfrac{1}{M})^{n\times M})^{T-1} = 0 \end{aligned}$$

In other words,

$$P(a^t = a^* \text{ eventually}) = 1 \quad \blacksquare$$

References

[1] Cardei M. and Du D.. 2005. Improving Wireless Sensor Network Lifetime Through Power Aware Organization, ACM Wireless Networks. Vol. 11, No. 3, pp. 333-340.

[2] Cardei M., and J. Wu., 2006. Energy-efficient coverage problems in wireless ad-hoc sensor networks, Computer Communications, Vol. 29, pages 413-420.

[3] Cardei, M. Thai M., Li Y. and Wu W., 2005. Energy-Efficient Target Coverage in Wireless Sensor Networks, IEEE INFOCOM 2005.

[4] Chen B., Jamieson K., Balakrishnan H. and Morris R. 2001. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks, Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, pp. 85-95

[5] Fudenberg D. and Levine D., 1998. *The Theory of Learning in Games.* MIT Press.

[6] Ganesan D., Cerpa A., Ye W., Yu Y., Zhao J. and Estrin D. 2004. Networking Issues in Wireless Sensor Networks, Journal of Parallel and Distributed Computing, Vol. 64 No. 7, pp.799-814.

[7] Garcia A., Reaume D. and Smith R.L., 2000. Fictitious Play for Finding System Optimal Routings in Dynamic Traffic Networks. Transportation Research B, Methods, Vol. 34 No. 2, pp. 147-156.

[8] Godfrey P., Ratajczak D. 2004. Naps: Scalable, Robust Topology Management in Wireless Ad hoc Networks, Proceedings of the third International Symposium on Information Processing in Sensor Networks.

[9] He T., Krishnamurthy S., Stankovic J., Abdelzaher T., Luo L., Stoleru R., Yan T., Gu L., Hui J. and Krogh B. 2004. Energy-Efficient Surveillance System using Wireless Sensor Networks, Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services.

[10] Lambert T., Epelman M., Smith R. L., 2005. A Fictitious Play Approach to Large-Scale Optimization. Operations Research. Vol. 53, No. 3, pp. 477-489.

[11] Man-Cho So A. and Ye Y. 2005. On Solving Coverage Problems in a Wireless Sensor Network Using Voronoi Diagrams, 1^{st} Workshop on Internet and Network Economics.

[12] Marden J.R., Arslan G., and Shamma J.S., 2005. Joint Strategy Fictitious Play with Inertia for Potential Games. 44th IEEE Conference on Decision and Control.

[13] Monderer D. and Shapley L., 1996. Fictitious Play Property for Games with Identical Interests. Journal of Economic Theory, Vol 68 No. 1, pp 258-265.

[14] Robinson J. 1951. An Iterative Method of Solving a Game. Annals of Mathematics, Vol 54 No. 2, pp. 296–301.

[15] Shamma J.S. and Arslan. G. 2005. Dynamic Fictitious Play, Dynamic Gradient Play, and Distributed Convergence to Nash equilibria, IEEE Transactions on Automatic Control, Vol. 50 No 3, pp. 312-327.

[16] Shapley, L. S. 1964. Some topics in two-person games. *Advances in Game Theory, L.S. Shapley, M. Dresher, and A. W. Tucker, Eds. Princeton, NJ: Princeton University Press*, 1–29.

[17] Slijepcevic S. and Potkonjak. M., 2001. Power Efficient Organization of Wireless Sensor Networks. In ICC 2001, Helsinki, Finland.

[18] Tian D. and Georganas ND. 2003. A Node Scheduling Scheme for Energy Conservation in Large Wireless Sensor Networks. Wireless Communications & Mobile Computing, vol.3, no.2, pp.271-90.

[19] Xing G., Lu C., Zhang Y., Huang Q. and Pless R. 2005. Minimum Power Configuration in Wireless Sensor Networks, Proceedings of the 6th ACM International Symposium on Mobile Ad hoc Networking and Computing.

[20] Xing G., Wang X.; Zhang Y., Lu Ch., Pless R. and Gill C. 2005. Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks, ACM Transactions on Sensor Networks, Vol. 1 (1) pp 36-72

[21] Yan Y., He T., and Stankovic J., 2003. Differentiated surveillance for sensor networks. First International Conference on Embedded Networked Sensor Systems, pp. 51–62.

[22] Young H.P., 1998. Individual Strategy and Social Structure : An Evolutionary Theory of Institutions. Princeton University Press.

[23] Zhang H. and Hou. J. 2005. Maintaining Sensing Coverage and Connectivity in Large Sensor Networks. Ad Hoc & Sensor Wireless Networks, March, pp. 5-17.